



Haga lo que hicimos en el aula: Checked e Unchecked

1) Primero, creamos la clase *MiExcepcion* *que hereda de *RuntimeException*:

```
public class MiExcepcion extends RuntimeException {  
  
}
```

[COPIA EL CÓDIGO](#)

2) En la clase *MiExcepcion*, crearemos un constructor que recibe un mensaje de tipo *String* y se lo pasa al constructor de la clase padre *RuntimeException*:

```
public class MiExcepcion extends RuntimeException {  
    public MiExcepcion (String msg) {  
        super(msg);  
    }  
}
```

[COPIA EL CÓDIGO](#)

3) Ahora lanzaremos nuestra excepción dentro de *metodo2()*, en la clase *Flujo*. Reemplacemos el *throw* actual con lo siguiente:

```
throw new MiExcepcion("Surgió un error");
```

[COPIA EL CÓDIGO](#)

4) También necesitamos agregar el tipo *MiExcepcion* dentro del *catch* en la clase *Flujo*:

```
try {  
    metodo1();  
} catch (ArithmeticException | NullPointerException | MiExcepcion ex) {  
    String msg = ex.getMessage();  
  
    System.out.println("Exception " + msg);  
    ex.printStackTrace();  
}
```

[COPIA EL CÓDIGO](#)

5) Ahora cambie la clase *MiExcepcion* para extender la clase *Exception* (dejando *checked*):

```
public class MiExcepcion extends Exception { //checked  
  
}
```

[COPIA EL CÓDIGO](#)

6) En la clase Flujo, haga que el código se compile nuevamente y use la excepción *MiExcepcion* en el *metodo1()* y *metodo2()*:

```
private static void metodo1() throws MiExcepcion {  
    System.out.println("Inicio del metodo1");  
    metodo2();  
    System.out.println("Fin del metodo1");  
}  
  
private static void metodo2() throws MiExcepcion {  
    System.out.println("Inicio del metodo2");  
    throw new MiExcepcion ("Surgió un error");  
}
```

```
// System.out.println("Fin del metodo2");  
}
```

[COPIA EL CÓDIGO](#)